Andrew Hautau

Deriving Knowledge From Data At Scale

Assignment #1, Report

The data set I’ve chosen for the time-series assignment consists of daily stock exchange returns from the Istanbul Stock Exchange (now under the [Borsa Istanbul)](http://en.wikipedia.org/wiki/Borsa_Istanbul), the SP 500, and stock market returns from Germany, UK, Japan, Brazil, the MSCI European Index, and the MSCI Emerging Markets Index. The data set is available from [the Machine Learning Repository](http://archive.ics.uci.edu/ml/datasets/ISTANBUL+STOCK+EXCHANGE).

Since writing the “checkpoint” report, I rehashed my objectives with the assignment after doing more reading on time series analysis in R, both assigned and on my own. In particular, two readings were of most help in providing focus to the assignment, namely [the Little Book of R for Time Series](http://a-little-book-of-r-for-time-series.readthedocs.org/en/latest/) and a [blog post at Rob Hyndman’s blog, Hyndsight](http://robjhyndman.com/hyndsight/tscvexample/). I did not move on to a multivariate analysis that integrated the other stock index time series, but I did achieve progress towards univariate analysis using the ISE index. The ISE time series will be the focus of my analysis here.

To begin, my choice of this data set was, for the most part, arbitrary. After my choice to use this data, it occurred to me that it was inherently irregular— being daily financial returns that had gaps on weekends, holidays, etc. To account for irregularity, I wrote a short helper function, ensure.regularity(), to coerce regularity and approximate all time intervals with “NA” values, which I depended on a good deal in my analysis. Additionally, I found the “zoo” package very helpful in providing assistance with dealing with an irregular time series.

The ISE time series is a time series of daily changes, and is already in a stationary state. Using the ur.kpss() function from the “urca” package, the summary gives a low KPSS level of 0.2995 for reg.ise, the variable for our stationary set (with coerced regularity). Additionally, we see that the ISE time series has some autocorrelation up to lag 3— based on reading the Little Book of R, I understand that this would inform a manual construction of an ARIMA model by helping choose p,d, and q variables. Moving on, visually we can confirm stationarity in the autocorrelation plots returned by plot(stat.kpss.test) and lag.plot(reg.add.ise). By contrast, the additive transformation of the time series has a high amount of autocorrelation, as indicated by ur.kpss(reg.add.ise).

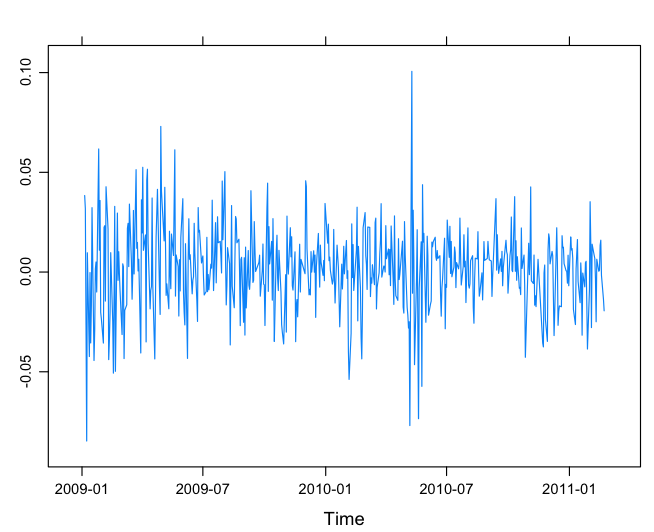
In cross.validation(), I’ve put together an iterative time series cross-validation that tests linear regression, ARIMA, ETS, and the Holts Winters forecasting models with horizons of 30 and a training window (K values) of 60, 20, 10, 5, and 4 in five different cross-validations (plots can be seen by running plot.cross.validation(ise.cv.list$mae) ). Not surprisingly, with higher k values, cross validation shows no single model drastically out-performing another when applied to the stationary time series. With decreasing k values, performance improves with decreasing mean average errors across all models, but significant performance differences amongst the model forecasts are revealed. Linear regression is our clear loser, with higher mean average errors than all others as the horizon of the forecast increases. Holt Winters, by contrast, is a winner in our cross-validation with the lowest K value (4), with MAE values that stay very flat relative to

When looking at forecast residuals for the stationary time series, the output of slide.forecast.errors(ise.cv.list)(a convenience function that iterates the plotForecastErrors()included within The Little Book of R Time Series), follow relatively normal distributions when the cross validation has a higher K value/variable (the size of the of the training set the model uses to predict). As K decreases, we see sharp spikes in residuals approach zero.

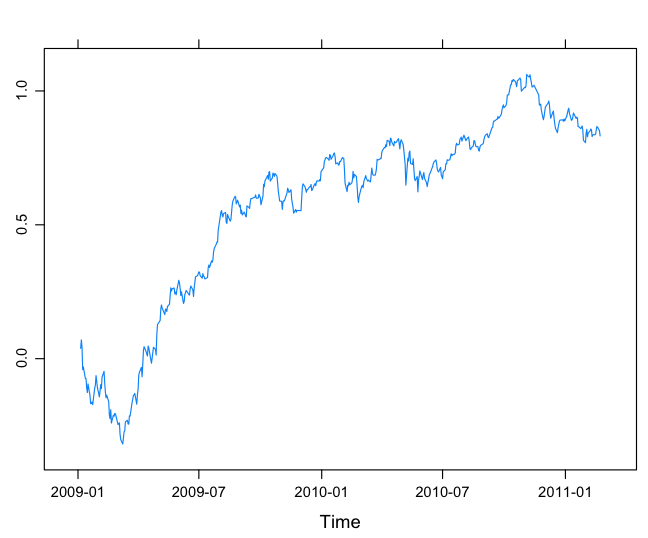
When applied to the additive time series, stark differences in forecast residuals are apparent. For all forecasts, residuals drift away from a normal distribution— an outcome of the autocorrelation demonstrated early for the additive time series by ur.kpss(reg.add.ise). This underlines the importance of stationarity in forecasting for time series analysis.

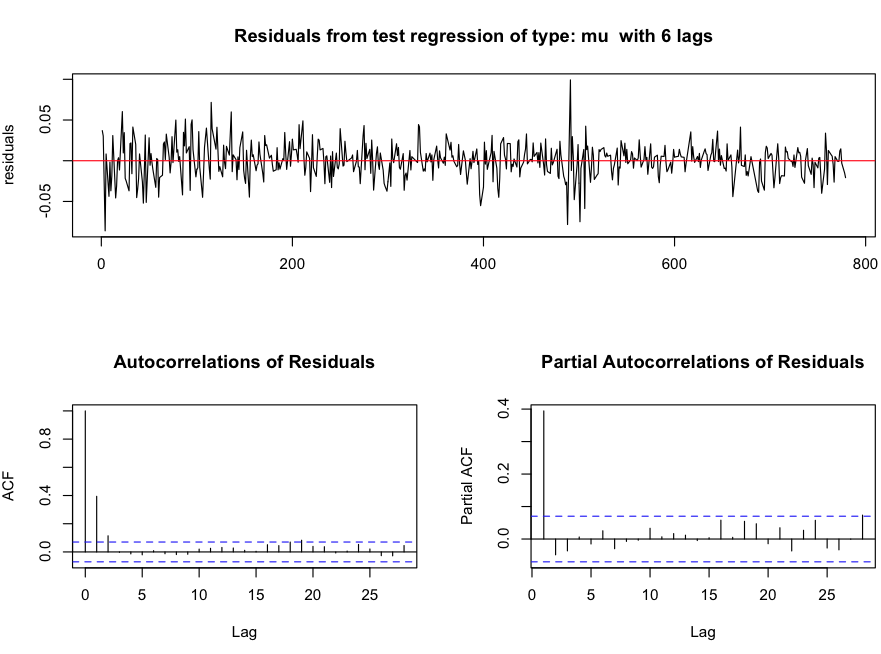
My analysis so far certainly has room for improvement. Unfortunately, the irregular nature of the data set has so far presented difficulty in getting at any potential trend or seasonal decompositions, which could further improve model performance. Still, this seems like a good starting point for developing a predictive model for daily financial returns and served as a decent primer for time series corss validation. In doing more with this particular data set, it would be exciting to involve the other stock indexes in developing forecasts.

Stationary ISE Time Series



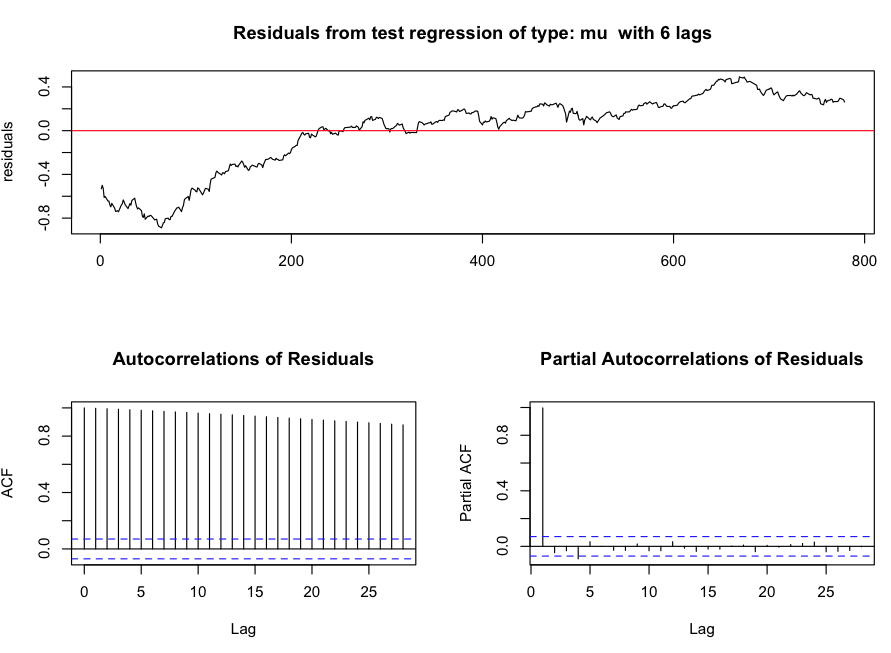
Additive ISE Time Series



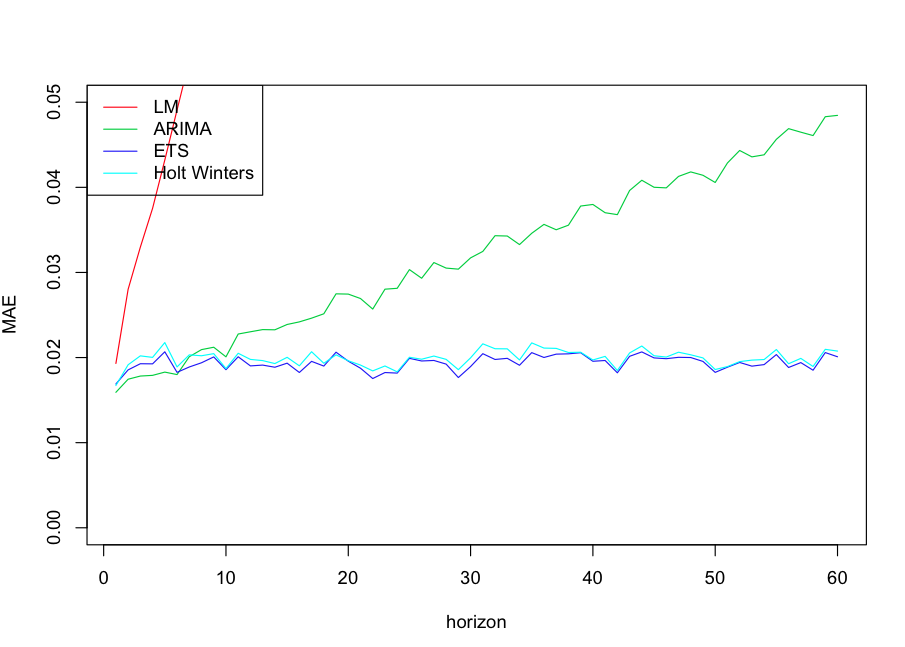
KPSS Unit Root Testing of Stationary Time Series

KPSS Unit Root Testing of Additive Time Series

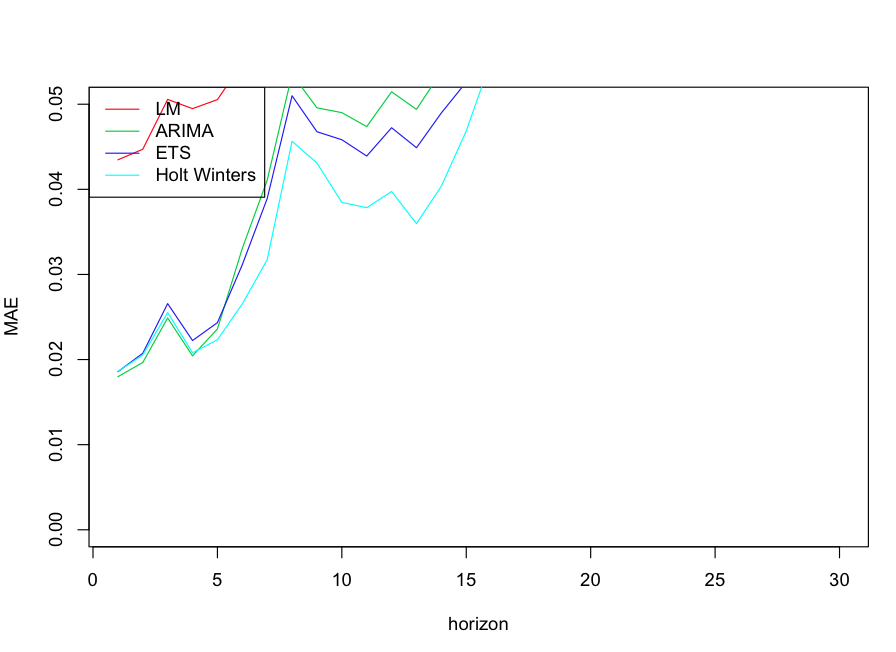
(Lots of autocorrelation!)



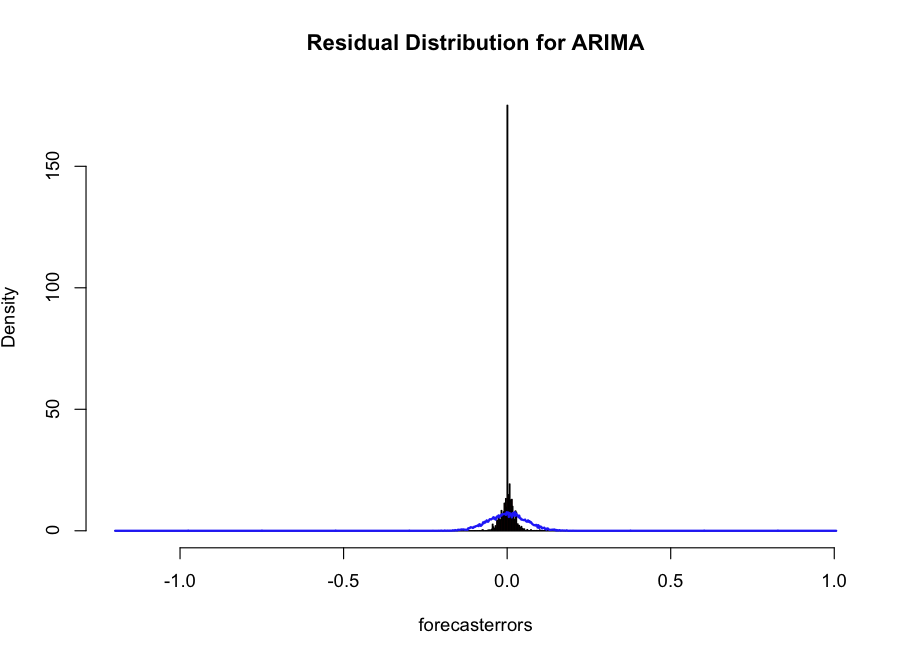
Cross Validation of Models for ISE Stationary Time Series, K=4, Forecast Horizon of 30



Cross Validation of Models for ISE Additive Time Series, K=4, Forecast Horizon of 30



Residual Distribution, ARIMA Forecast for Stationary ISE Time Series, K=4, Forecast Horizon of 30



Residual Distribution, Holt Winters Forecast for Additive ISE Time Series, K=4, Forecast Horizon of 30

